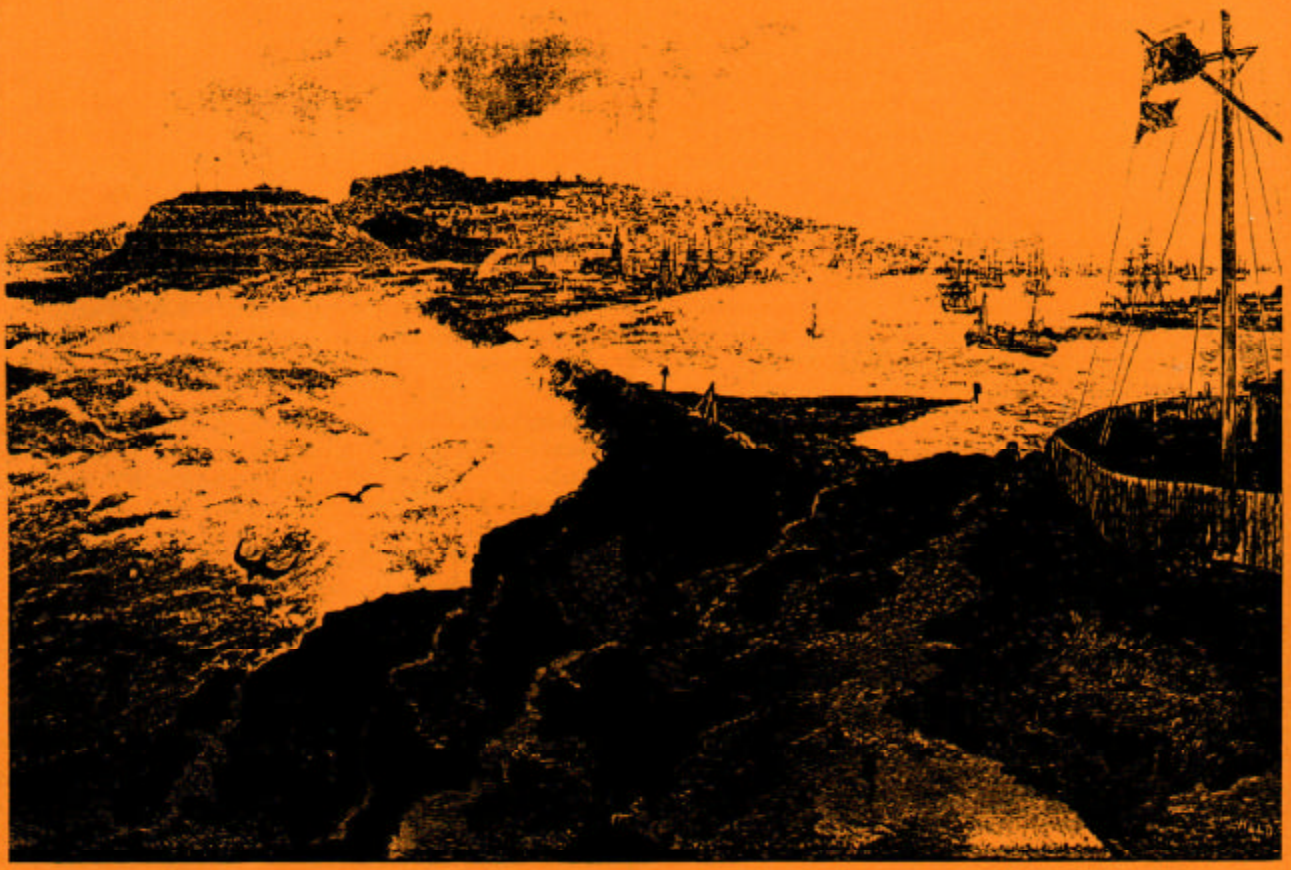
HUNTER VALLEY 99ERS USERS GROUP

HOME COMPUTER NEWSLETTER



"Newcastle from Nobby's Head" from *Picturesque Atlas Of Australasia, 1888*

NOVEMBER 1989

REGISTERED BY AUST POST
PUBLICATION No. NBG8023

YOUR COMMITTEE

PRESIDENT

Peter Smith
8 Glebe St.,
EAST MAITLAND 2322
Phone 336164 V/tl 493361640

VICE PRESIDENT

John Paton
1 Parien Close,
RUTHERFORD 2320
Phone 326014 V/tl 493260140

SECRETARY

Brian Woods
9 Thirlmere Pde.,
TARRO 2322
Phone 662307 V/tl [illegible]

TREASURER

Noel Cavanagh
378 Morpeth Rd.,
MORPETH 2321
Phone 333764

Software Librarian

Stewart Bradley
14 Hughes St.,
BIRMINGHAM GARDENS 2287
Phone 513246

EDITOR

Allen (Joe) Wright
77 Andrew Rd.,
VALENTINE 2280
Phone 468120

PURCHASING/HARDWARE CO-ORDINATOR

Alan Franks
822 Pacific Highway,
MARKS POINT 2280
Phone 459170

SOCIAL SECRETARY

Robert (Bob) MacClure
75 Deborah St.,
KOTARA SOUTH
Phone 437431

PUBLICATIONS LIBRARIAN

Ken Lynch
9 Hall St.,
EDGEWORTH 2285
Phone 585983

COMMITTEE MEMBERS

Don Dorrington
36 NELSON St.,
BARNSLEY 2301
Phone 531228

Tim Watkins
36 The Ridgeway,
BOLTON POINT 2283
Phone 592836

CONTRIBUTIONS

Members and non members are invited to contribute articles for publication in HV99 NEWS.

Any copy intended for publication may be typed, hand written, or submited on tape/disc media as files suitable for use with TI Writer (ie. DIS/FIX 80 or DIS/VAR 80). A suitable Public Domain word processor program will be supplied if required by the club librarian.

Please include along with your article sufficient information to enable the file to be read by the Editor eg. File Name etc. The preferred format is 35 columns and page length 66 lines, right justified.

All articles printed in HV99 NEWS (unless notified otherwise) are considered to be Public Domain. Other user groups wishing to reproduce material from HV99 NEWS may feel free to do so as long as the source and author are recognised.

Articles for publication can be submitted to the Editor, ALL other club related correspondence should be addressed to The Secretary.

DISCLAIMER

The HV99 NEWS is the official newsletter of the HUNTER VALLEY NINETY NINE USER GROUP.

Whilst every effort is made to ensure the correctness and accuracy of the information contained therin, be it of general, technical, or programming nature, no responsibility can be accepted by HV99 NEWS as a result of applying such information.

The views expressed in the articles in this publication are the views of the author/s and are not necessarily the views of the Committee, Editor or members.

TEXAS INSTRUMENTS trademarks, names and logos are all copyright to TEXAS INSTRUMENTS.

HV99 is a non profit group of TI99/4A computer users, not affiliated in any way with TEXAS INSTRUMENTS.

random bytes

with

BOB CARMANY

Another month and time for another column --always an interesting endeavor! Let's take a look at some XB programming tips this month.

I'm sure that everyone knows how to get those extra long lines typed in by using <FCTN-8> (REDO). and there have been several digressions in past issues regarding the use of the pre-scan. With judicious use, both can make a program execute faster and get as much into each program line as possible. Now it is time to look at some other tips to make a program more pleasing to the eye.

A couple of years ago, I did a review of a program released by Miller's Graphics called NIGHT MISSION. The program itself was well done but the best part of the package was the manual that came with it. Besides a complete flowchart of the program, there were numerous tips and tricks that could be used by the novice programmer.

One of the best ideas was at the very start of the program. Nothing is more boring than to see a program start and see graphics presented in a piecemeal fashion on the screen. You know, random generation of a starfield---watching the stars appear one by one isn't the most exciting way to spend a couple of minutes. Craig Miller and Co. used a much better aproach. The program started off with a title screen with a sprite zooming across it --in this case a helicopter. Taking advantage of the fact that once a sprite is set in motion it needs no further control, our attention is diverted while we watch it and listen to the sound effects. While we are momentarily distracted, the program initializes and places a starfield on the screen. Although the characters are there, they are not visible because of one simple fact---the character sets containing them have not been initialized with CALL COLOR. With everything in place, a CALL COLOR causes the entire starfield to appear at once as if by magic. A truly nice touch. There are all sorts of variations that can be used with sprites and character definitions.

Another interesting idea is "layering" sprites. A sprite's "hierarchy" is determined by its number. Higher numbered sprites can be placed on top of lower numbered sprites --sort of like a sandwich. For example, SPRITE #1 could be "layered" beneath SPRITE #2. The result would be that #2 could pass over #1. If the motion parameters are zero (making #1 stationary) and number #2 had empty space in it, you could see #1 as #2 passed over it. Using stationary sprites instead of characters, you could have 3-dimensional displays with sprites passing in front of or behind other sprites. Think about that one for awhile!!

Here is a modified version of a program by W.K. Baltrop to illustrate layering sprites.

```
100 CALL CLEAR :: CALL SCREEN(2)

110 CALL CHAR(96,RPT$("F",64))

120 CALL CHAR(100,"8060307F7F30608"
RPT$("0",48))

130 SP=2 :: CALL MAGNIFY(4)

140 FOR X=150 TO 21 STEP -16

150 CALL SPRITE(#SP,96,SP/2+2,X,200
-X)

160 SP=SP+2 :: NEXT X

170 SP=1

180 FOR X=150 TO 21 STEP -16

190 CALL SPRITE(#SP,100,16-SP/2,X+8
,200-X)

200 SP=SP+2 :: NEXT X

210 FOR X=1 TO 27 STEP 2

220 CALL MOTION(#X,0,5):: NEXT X

220 CALL KEY(0,K,S)::IF S=0 THEN
220 :: IF K=32 THEN CALL DELSPRITE(
ALL):: CALL SCREEN(8):: GOTO 100

230 END
```

Brian Woods REPORTS From the Secretary's ** Desk **

NEWS FROM THE STATES

From the October issue of Ildbits, the newsletter of the Mid South Users Group comes the following items from Gary Cox's column...

"Texaments of 53 Center Street Patchogue, NY 11772 has released TI Artist Plus! The new version of TI Artist called TI Artist Plus! is said to have a total of six individual modules within the program, including a drawing module, enhancement module, vector module, font module, print module & movie module. Among the many features is the movie module allowing the creation of short animated sequences... With the print module 1 to 3 pictures can be printed simultaneously... The vector module can be used to scale your drawings... In all TI Artist Plus! looks full of new features & the old version of TI Artist was great as it was! Current owners of TI Artist can upgrade by sending the original TI Artist disk & the front page of the original manual along with $US14.95 + $US2.50 shipping. New purchasers can purchase TI Artist Plus! for $US24.95 plus $US2.50 shipping."

"Also new from Texaments is TI BASE version 2.02 where some minor bugs have been corrected and hard drive support has been added. Owners of TI BASE 2.0 and higher may upgrade to 2.02 by sending the original disk + $US2.50 shipping. Earlier versions of TI BASE (before 2.0) can upgrade for $US7.95 + $US2.50 shipping."

"Press, according to rumour, is supposed to be released at the Chicago TI Faire, nearly one year after its announced release."

From the September issue of the Chicago TImes newsletter comes the following from Bill Gaskill's column "Four-A/Talk"...

"Word is that Warren Agee is readying the next version of First Base and Genial Computerware is looking for suggestions from users on new features that they would like to see in it. You may write to Peter Hoddie at the Genial Computerware address with your suggestions."

"J Peter Hoddie has been working on a program tentatively named Sign Shop that will operate like Broderbund's Print Shop for Apple and IBM type computers. JPH also intimated that he hopes to produce a Naverone to First Base conversion program to allow owners of the Naverone DBMS to port their data files over to First Base format."

THE LIGHTNING SEASON

From the November issue of the Sydney User Group's Newsletter comes this timely warning...

"The summer season is fast approaching and with it the violent summer electrical storms. Your computer and modem are particularly susceptible to serious damage from lightning induced surges which enter via either the power mains or the telephone line. Even when the telephone line is underground it can still convey lightning surges into your modem."

"The best advice that I can offer to protect against lightning surge damage is to UNPLUG both the telephone line and the power cord from your modem and computer when a storm is approaching. This does little to protect against the unexpected surge however. Telecom can fit a lightning suppressor - for a price - but it is not guaranteed that it will definitely protect against a very near lightning strike. The Telecom suppressor is a gas arrester which connects to each

leg of the line and to a Telecom earth. Mains surge suppressor filters can be purchased to protect against surges on the power lines."

Following on this article, Ross added the following 'extra'...

"I have just become aware of a new product which will provide protection from surges, both on the power mains & the telephone line. The unit, which is named Faxguard, plugs into both the 240V power mains & the telephone socket. The computer equipment (or Fax machine) plugs into surge protected modem and power outlets on the Faxguard..."

"Faxguard is manufactured by Critec Pty Ltd of Hobart & may be purchased from Component Resources (NSW) Pty Ltd of Girraween NSW phone 02-688 4528. Price quoted is $195 + sales tax."

COMPUTER RELATED HUMOUR

From the September/October issue of the newsletter of the Forest Lane User Group comes this piece on computer programmers...

"One who passes himself off as an exacting expert on the basis of being able to turn out, after innumerable debugs, an infinite series of incomprehensible answers calculated with micrometric precision from vague assumptions based by persons of dubious reliability and questionable mentality for the purpose of annoying and confounding a hopelessly defenceless department that was unfortunate enough to have asked for the information in the first place."

Naturally the hardware itself is often a target of the humourist's wit. Take for example this article from the November issue of the East Anglia Region Users Group in the UK...

"This machine is liable to break down when you need it most. A special sensor has been fitted to this machine, enabling it to detect when it is being used for critical operations of supreme importance to the operator. It then waits for a random period of time and breaks down for no apparent reason. Nobody knows how this is done. IT JUST KNOWS!!"

"UNDER NO CIRCUMSTANCES whistle carelessly as you approach the machine. This is a dead giveaway."

"Never threaten the machine with violence. This merely aggravates the situation and can lead to personal injury to you when the breakdown finally happens."

"Never try to use another machine. They can communicate with one another, so this could lead to mass breakdowns throughout the building."

"Never let anything electronic know you are in a hurry!"

IT'S CHRISTMAS!

Well folks, it's that time of the year again. It seems that the older we get the faster time flies! It seems like no time at all has passed since the last time the decorations were dragged out and dusted off, the Christmas sales started & the kids started nagging about the latest in skateboards, bikes etc that they could not possibly live without! Oh well, time waits for no man.

I would like to wish all our members, their families & our readers both in Australia & around the world all the best for Christmas & that you have a happy, healthy & prosperous New Year.

MMMmmmmm. From Joe!

Don't forget the deadline for the

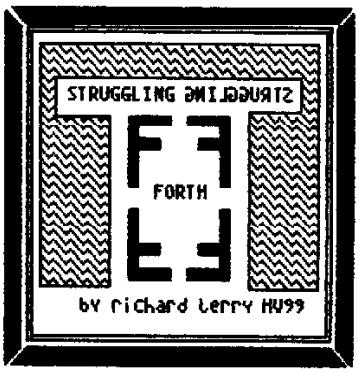
CHRISTMAS BUMPER ISSUE !!!!

SCR #200

STRUGGLING FORTH

HV99ERS NOVEMBER 89 ARTICLE

TI FORTHPRINT RELEASE



Rather than continue with our editor program development this month, I thought I'd formally release my print,utility program, which Joe and I use extensively in program development. Joe actually pushed me into re-writing some of my programs to release in Ti-forth. Next month I will release my Forth< >files program which I have just completed, and hopefully then resume the series on program writing in the new year. Unfortunately it has been unseasonally busy and I've had little time for computing in the last few weeks. I thought the easiest way to describe the program would be to reprint the program documentation.

FORTHPRINT by Richard Terry (Version 1.0)

INTRODUCTION

FORTHPRINT is a simple forth screen print utility for use in the TI-FORTH environment to produce a screen printout consisting of 6 screens of code or text-mode graphics images per page. It has been designed to be user friendly, with most options being self explanatory.

FORTHPRINT requires the following minimal TI-99/4A system configuration:

- TI-99/4A console with monitor or TV set
- One disk drive storage system
- 32K expansion memory system
- Editor/Assembler module or
- Extended Basic module

A printer is obviously necessary to produce the output!!!

FEATURES

-Printing out forth screens from any forth version.
-Printing out text-mode graphics images either as a
 listing on forth screens or as a seperate screen dump
 in either normal, near letter quality, or condensed format
-viewing of either type of screen
-rebooting of any Ti-forth disk
-Installation in your system to any forth screen

LOADING THE PROGRAM

Using the Editor/Assembler's Option 3, type "DSK1.FORTH", or if using an Extended basic module, it will auto-load after choosing the extended basic from the title screen. The program will execute automatically and present you with a master menu. This uses the now familiar "bar-menu" style. To choose your option use the E or X keys (not the Function E/X), and press ENTER at your selection.

THE MAIN MENU SELECTION.

EDIT

On choosing either SOURCE CODE or WINDOW/DOC you will be presented with a screen of default options to edit. The bar menu at this level is horizontal and will respond to the S or D keys, (not the Function S/D), and ENTER to run your option. You may after editing, prior to a printout, use the VIEW option to browse through your disk, viewing either source code in identical format to your usual forth editor, or window code, displayed as if you were watching your TV screen. In the WINDOC/DOC viewing mode you can dump the screen contents to printer in either normal, normal + near letter quality, or condensed print for use elsewhere.

A special note on Validate scrn

Some forth screens you encounter may have bytes present which when emitted to the printer are equivalent to an actual printer control code, hence they do bizarre things like do a form feed, reverse your printer, make the print expanded etc. Answering Y to validate is only necesary once you encouter the problem. Do not leave on all the time as checking each screen prior to printing slows things up.

CONFIGURATION

This segment has been included as I have found that not all printers have the same character values corresponding to the graphic character component of forth - ie the character set you have allocated to your text-mode graphics. After updating you can permanantly save the binary image of your program back to disk, or if only using on a one-off basis, use now but not save. If you do not have the original distribution disk in drive 1 the program will not save the configuration.

DOCUMENTATION

Two quick help screens are included as a an on-screen overview and their contents are self explanatory.

RE-BOOT

This interposes a pause between the user and the re-boot. You may insert any Ti-forth compatable disk into drive 1 and re-boot.

GENERAL COMMENTS.

This program was developed for my own use, especially the segment to allow the dumping of text-mode graphics to the printer, either independently or displayed on a forth screen listing similar to source code, except in a 40/24 format. Many forth programmers will not use a similar method of displaying their text-mode graphics displays.

There are several other programs in this series entitled "Forth Programming Utilities". The others, which have been written but not as yet released include a text-mode full screen design editor, a program to build individualised pattern descriptor tables, a forth style disk manager, and a forth<->files program. These ideas are not new and the code to perform these functions has been published in many a user group magazine. I found it useful to write an entire program to perform these functions because it speeded up non-utility related program development.

A few quirks of Forthprint. Hitting Funtion 8 at the **beginning** of an input will jump you back to the previous screen, equivalent to the F9 of most programs, but being a touch typist I find the former combination falls easier to hand. Secondly in the horizontal bar-menu and some other situations where there is no apparent escape if you've made a mistake try the good old fashiond F9 key! Parts of the configuration screen are probably unnecessary, it is an old routine I've never changed; I may alter it if any feedback is received.

The ability to view screens allows you to browse through the disk when you can't quite remember which screens you want to print after you've booted up!, and as a bonus allows a screen dump of any graphics screens to the printer. Please note this is not a true screen dump as the inverted portions are printed non-inverted.

FOR FORTH PROGRAMMERS

A few notes should you find this program worthy of incorporation into your own system, as opposed to always having to boot it from its own disk.

It is written in Ti-forth, I usually program in Superforth, so I have the codings available for either version, the differences being those of address overlays. Decompilation of the code would reveal it to be not particularly clever. I have made no attempt to optimize the code for speed of execution or elegance of design. In fact it is probably "bad" forthcode.

The text-mode graphics screens on the disk are not needed for program execution, as these screens are also contained in the dictionary for simplicities sake as programming space was not a problem. I have left them on the disk to illustrate the sort of displays one can use in a forth program without needing code to produce the display. When not kept in the dictionary, no program space is lost. Typical images can be generated in a few minutes using a full screen design editor or your own writing, or from my own, when I get around to releasing it. The two screens of pattern descriptor tables are needed for program visual displays.

Installation.

Within the configuration option, by choosing F5, you may install the program on any disk or screen in your system, and boot it from that location when needed. The installation routine writes a boot screen, shifts the two pattern descriptor tables and updates the load information for them, as well as moving the documentation screens and dictionary - needing a total of 21 screens in all.

Copying the distribution disk.

As forth needs to access screen 3 to boot, and the sectors containing the program must be kept as they are, you may not file-copy the disk. It may be copied by most disk managers using either sector copy or bit-map copy. The forth programs are contained as true D128 files, however the bitmap of the disk has been altered to "hide" sectors 3->9 from use by the disk manager.

DISK ORGANISATION.

Here is the disk organisation by block:

```
    0 RESERVED FOR FILENAMES AND BIT MAP
    1 RESERVED FOR FILENAMES
    2 RESERVED FOR FILENAMES
    3 FORTHS REQUIRED BOOT SCREEN
  4-5 ERROR MESSAGES
  6-7 EMPTY
    8 EMPTY
    9 COMMENT IDENTICAL TO THIS RE DISK ORGANISATION
   10 BOOT SCREEN FOR THE PROGRAM
11-12 NORMAL AND INVERTED PATTERN DESCRIPTOR TABLES
13-14 DOCUMENTATION FOR THE PROGRAM
15-30 BINARY IMAGES OF FORTHPRINT
31-34 GRAPHICS IMAGES USED IN COMPILING THE PROGRAM**
35-ON FORTH, FORTHSAVE, FPRINT/DOC, EMPTY SCREENS.
```

**Note:these window screens are not necessary for the program, they are included to allow the user unfamiliar with this type of screen display to peruse them using the program.

THIS IS A FAIRWARE PROGRAM

and is distributed in the spirit of fairware. If you find this program useful your contribution can be sent to the author below, along with suggestions for improving the program, or to report any "bugs".

RICHARD TERRY
141 DUDLEY RD
WHITEBRIDGE 2290
NSW AUSTRALIA.
049 436861/436511

As examples I will show below both types of screen output:

Program: Demo-textgraphics Date: 23Nov89

Screen # 9

```
        Forth programming Utilities

                                    Fairware

  Choose option                     Written by
                                  Richard Terry
  Design editor                   141 Dudley Rd
  Auto-character                  Whitebridge
  Configure disk                    Nsw 2290
  Documentation                   Australia.
  Re-boot disk
  Quit Program                      Hv99ers

  TM-editor  Super4th version 1.0  11/89
```

Screen # 10

```
                Auto character pattern generator

                012345  HEX   As printed  Save/load

                0                         Char# :32
                1                         PDT   :N
                2                         To Scr:823
                3
                4                         Quit
                5                         Redo
                6                         Accept:A
                7

                Current pattern descriptor table

  F4/F6 view PDT     F9     quit editor
  F5 save pattern    /   Bar pixel on/off
  F7 load pattern  Plus usual editor keys
```

Forth Utilities V1.0 - Forthprint - <c> Richard Terry 1989

Program: Demo-source code Date: 23Nov89

Screen # 40

```
\ Ti-forthprint  System calls  RHT9Aug89

: VSBW 0 SYSTEM ; : VMBW 2 SYSTEM ; : VIOR 26 SYSTEM ; : VSWR 4
SYSTEM ; : VMBR 6 SYSTEM ; : VWTR 8 SYSTEM ; : CLS 16 SYSTEM ;
: VFILL 20 SYSTEM ; : TRL -TRAILING ; : DSRLNK 8 14 SYSTEM ;
: AT GOTOXY ; : DEC DECIMAL ; : PAGE CLS 0 0 AT ; : VAR VARIABLE
 ; : CON CONSTANT ; : EBS EMPTY-BUFFERS ; : 2DUP OVER OVER ;
: MON 32 10 SYSTEM ; : NDROP 0 DO DROP LOOP ; : 2IROP 2 NDROP ;
; BSAVE FLUSH BEGIN SWAP >R DUP 1+ SWAP OFFSET @ + BUFFER UPDATE
 DUP B/BUF ERASE R OVER ! 2+ HERE OVER ! 2+ CURRENT @ OVER ! 2+
LATEST OVER ! 2+ CONTEXT @ OVER ! 2+ CONTEXT @ @ OVER ! 2+ VOC-L
INK @ OVER ! 2 + 29801 OVER ! 10 + HERE R - R> DUP 1000 + >R SWA
P >R SWAP R> 1000 MIN CMOVE R SWAP HERE R> < UNTIL SWAP DROP
FLUSH ; : L>U DUP 96 > IF 32 - THEN ; : 3DROP 3 NIROP ; : ->VDU
BLOCK 0 960 VMBW ; : 5DROP 5 NDROP ; : PICK 2 * SP@ + @ ; : NDUP
 DUP 0 DO DUP >R PICK R> LOOP DROP ; : 3DUP 3 NDUP ; : 2OVER SP@
 6 + @ SP@ 6 + @ ; : <> = 0= ; : P PICK ;
```

Screen # 41

```
\ Ti-forthprint  string wrds  RHT9Aug89

: (!") R COUNT DUP 1+ =CELLS R> + >R >R SWAP R> CMOVE ;
: !"   34 STATE @
       IF COMPILE (!") WORD HERE C@ 1+ =CELLS ALLOT
       ELSE WORD HERE COUNT >R SWAP R> CMOVE THEN ; IMM
: HCHAR  >R >R SCRN_WIDTH @ * + SCRN_START @ + R> R> VFILL ;
\ compares two strings at adr1, adr1 leaves flag
: SAME$ COUNT ROT COUNT ROT MAX 1 SWAP 0     ( Adr1 Adr2--flag )
        DO >R DUP C@ ROT DUP C@ ROT =        ( Where these adr )
           R> MIN SWAP 1+ ROT 1+ ROT         ( contain strings )
        LOOP >R DROP DROP R> ;               ( with counts     )
\ takes segment of string at  adr 1 puts in adr 2
: SEG$  ROT DUP >R OVER SWAP C! SWAP ROT + R> ROT 0
        DO OVER OVER SWAP C@ SWAP 1+ C! 1+ SWAP 1+ SWAP
        LOOP DROP DROP ; ( From adr, adr to put, start,num--)
: DELAY 0 DO NOP LOOP ;
```

Bits and Pieces with Joe Wright

Whether we, as individuals or groups, move through life we can offend quite unintentionally those whom we wouldn't want to. Unintentional or not offence can still be a stinging nettle to the receiver. This can then prompt an even more stinging reply and so on, somebody has to break this chain of events. In such a situation the words of Dickens seem apt.

May I tell you why it seems to me a good thing for us to remember wrong that has been done us? That we may forgive it.

DICKENS.

NEW DRIVES.

Bev Warren one of our out of Town members, she lives in Broken Hill took advantage of the cheap disc drives Alan Franks found. She tells me that the pair of slimlines are working just fine and are really quite. While talking to Bev we got onto the topic of children using the TI, she was looking for a simple but effective word processor for her children. They were having trouble with the Editor and it's complexity. I personally love the TI Writer Editor, but can understand that children could have difficulty with it.

This started me thinking about this problem and decided to have a hunt through my library to see what I had. A short rummage through the disc boxes, found IT! just what the doctor ordered. A simple word processor from the old Home Computer Journal, it is called HCJ WORD.

I have looked at this programme years ago and dismissed it as useless. After all TI writer is great. But when viewed with a new set of criteria, it now looked well suited to the task required of it.

The programme allows a page of text to be typed into the computer, 80 characters wide and 60 lines long. The line and character count is indicated in the top left corner of the screen as one types across the page. Apart from the normal function keys for insert, erase etc only five other key strokes are available;

CRTL 1 --- SAVE TO DISC.
CRTL 2 --- LOAD FROM DISC
CRTL 3 --- PRINT TEXT
CRTL 4 --- ERASE TEXT

FCTN 9 --- END SESSION.

The save option saves to a disc in D/V 80 files. This allows the TI WRITER formatter to be used if required. Although I have not yet tried it I would suggst that dragon slayer could also be used to check spelling.

It can also be loaded into the TI Writer editor and patched by Mum or Dad if need be.

Each of the above keys are self explanatory. The editor has no word wrap, reformat ets. What you type is what you will get

WORD WEAVE.

While talking about the word processor from H.C. Journal I can't forego mentioning Word Weave. Word weave is one of the programmes on the set of two discs on which the word processor was released. I think that this is an excellent programme. It allows the user to write a story in which the reader is given up to four options after each chapter. Selecting different options at the end of each chapter allows the reader to select the direction in which the story will wind.

It operates similar to those "Make your own story" books that you see in most book stores. They are aimed at young readers. I have demonstrated the programme at one of our meetings last year.

A COMPETITION.

Here is something for the kids to do over the christmas holidays. A writting competition. I would like our young members to write a story using Word Weave. You can develop it on paper first but the final draft which you submit for the competition must be on disc and typed in using word weave. Judging will be done at our March general meeting by all the members who attend. People who submit entries obviously won't be allowed to judge, neither will those whose children have entered. all others will. A minimum of 10 paragraphs must comprise the story. This is to allow the word weave to WEAVE. The prize or prizes will depend on the number of entries and their quality. I can tell you that I will donate the prize/prizes, it will be a book suitable for the age of the writer.

If you don't have Word Weave and want to enter the competition then contact me about how to obtain a copy.

If you can't type then by all means ask mum or dad to type your story in. But!! I am depending on your honesty to write it yourself without outside help from mum, dad or old brothers and sisters etc.

So happy writing, I look forward to getting some input. If space permits I will print all of the entries in the newsletter over a couple of months.

The following comes from the May 1988 L.A. Topics, written by Earl Raguse.

"The following programme called HIGHLIGHT makes permanent foreground/background color changes an can be controlled ON and OFF at will. Once executed, the programme can be deleted with NEW before you start entering a new programme. I some times put this in my LOAD programme, its easy to turn off if you don't want it.

I found the basic programme idea in the Tacoma ((ers Newsletter of December 1987, the article was by Joe Nolan, who credits Harry Wilhelm of the Twin TIers UG with the original idea. I don't have any idea how much evolution has gone on, but I added my two cents also.

Lines 130 and 140 do all the work, and if you wish to transfer this effect to one of your own programmes, thats all you need. The following tells how you can change these lines to suite your needs. If you study it a bit, you can see the potential for other purposes.

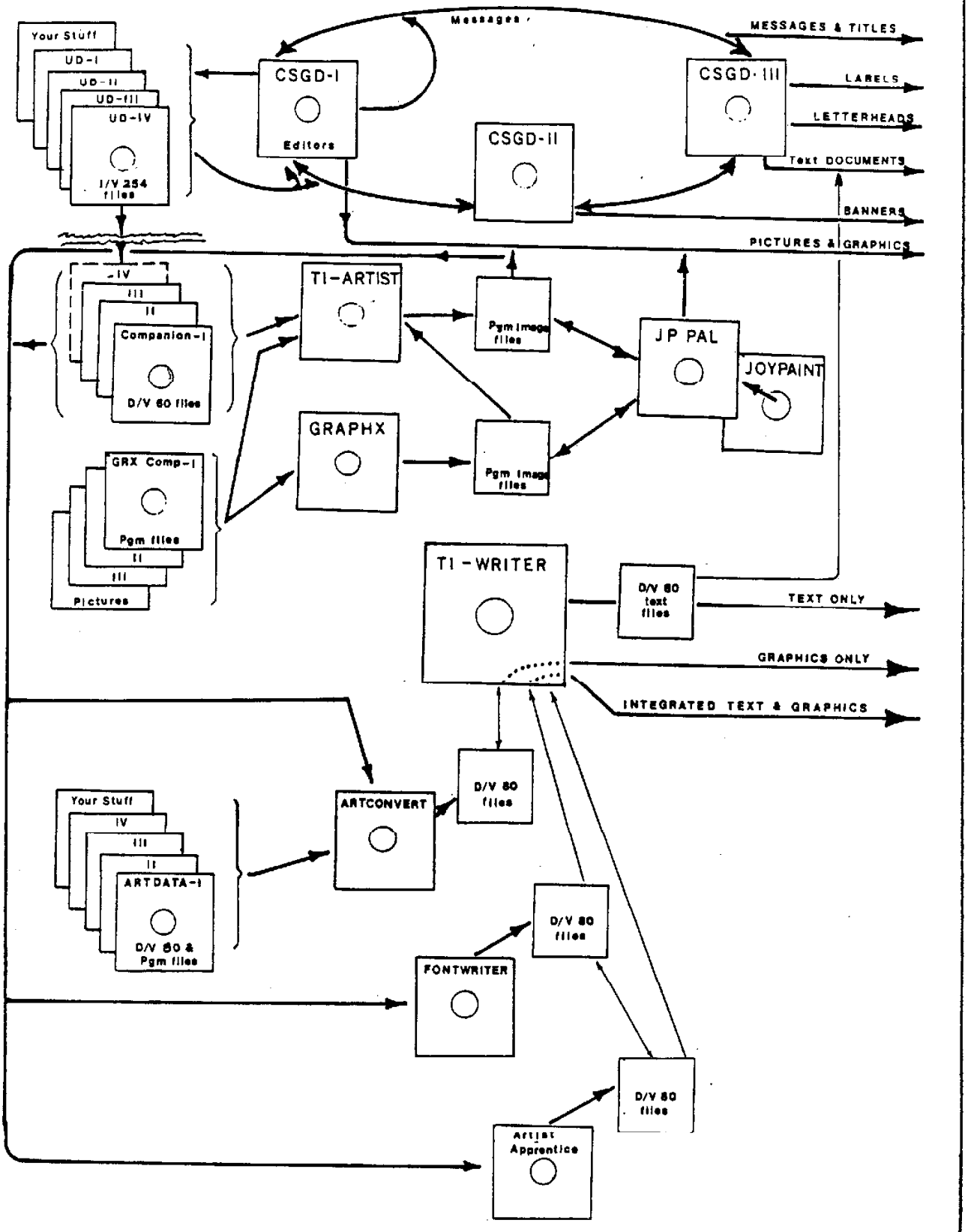
In line 130,
1) Change the eighth number, from the address, 17, to the number of the first character set you want to change PLUS 15. The current programme is 15+2=17 for character set 2.
2) Change the eighth number after that, 3, to the number of character sets to change. The current programme is for 3 for character sets 2, 3, 4.

In line 140,
1) Load a number, (in this case 244) for each character set to be changed. That number is computed as (16*(FG-1))+(BG-1) where FG and BG are the foreground and background colour numbers as defined in the XBASIC manual. Each character set could have a different combination of colours. The programme as written is for all characters white on blue, is (16*(16-1))+(5-1)=244.
2) The effect is turned ON by CALL LOAD(-31804,63) and OFF by CALL LOAD(-31804,0). This can be done either in a programme or from the keyboard. I added the lines 150 and 160 for easy control of the effect ON or OFF. These can be deleted if not wanted.

```
100 !SAVE DSK1.HIGHLIGHT
110 !By Joe Nolan, TACOMA 99ers
UG Newsletter Dec 1987,
Original idea by Harry Wilhelm of
TWIN TIers UG
120 !Modified by E.Raguse UGOC 1987
130 CALL INIT::CALL
LOAD(16128,2,224,38,0,2,0,8,17,2,1
,63,36,2,2,0,34,4,32,32,36,2,224,
131,192,3,128 )
140 CALL LOAD(16164,244,244,244)::
CALL LOAD(-31804,63)
150 PRINT "TURN IT OFF? PRESS SPACE
, ELSE ANY"
160 CALL KEY(0,K,S):: IF S=0 THEN
160 ELSE IF K<>32 THEN END ELSE
CALL LOAD(-31804,0)
```

CIN-DAY

SOME POPULAR GRAPHICS PROGRAMS



PUBLICATIONS
LIBRARY

The Library keeps getting bigger and bigger every month. So there are two possible solutions to the problem. All you TI readers out there are going to borrow more magazines or I will have to start to look for a new storage area. IF the first option fails ie. you don't borrow more magazines, then we will have to store them at my works.

THE AMBULANCE STATION
1 MAIN Rd.
BOOLAROO.

This is also the venue for this years Christmas party. The party will be mainly in the hall at the rear of the Station. It is owned by the Boolaroo Ambulance Ladies auxiliary. The Ladies have just stopped running housie days after fourty years of fund raising.

The hall will be passed on to the staff as a staff hall. The hall is right next to a vacant block, so noise is not a problem.

Well just a quick note on the high scores in nov 88 magazine. My son Bradley had a high score of 330,000 on Pinball. I couldn't even get close to it.
The magazine library not only consists of magazines from other TI-User Groups but several books on the TI.

Here is a list of some of the books.

ARTSGRAPHICS-T.THOMPSON,JR.
TI-PLAYGROUND-F.DIGNAZIO
STIMULATING SIMULATIONS-G.ENGEL
BASIC TRICKS-A.WYATT
TEXAS PROGRAM BOOK-V.APPS
101 TIPSTRICKS-L.TURNER
CREATIVE PROGRAMMING-L.STORM
CREATIVE PROGRAMING -ALL STARS
FUN GAMES-S.MUNCY
GAMES FOR KIDS-R.INGALLS
DYNAMIC GAMES-S.VINCENT
PROGRAMING BASIC-McGRAW-HILL
33 PROGRAMS-B.FLYNN
TEXTIGER II-F.McCARTY
PRINTER PERIPHERALS-AXIOM
FONT WRITER-P.HODDIE
TERRIFIC GAMES-RENKO/EDWARDS
CIRCUIT DIAGRAMS-TI
PROGRAMING FOURTH
USER'S REFERENCE GUIDE
PROGRAMMING GUIDE FOR SPRITES
PROGRAMMING AIDS III

But please don't forget these books are only bought to the AGM and December meetings. At monthly meetings i only bring the latest magazines for this year. All the other magazines that the club has are at my home. If you wish any others just give me a call.

THANK YOU
KEN LYNCH

EXTENDED BASIC

PROTECTION

Have a XB program that you wish to protect from listing. First do a CALL INIT in command mode then add this one (long) liner to the start of your program.

```
1 CALL PEEK(-31952,I,J,K,L)::
I=I*256+J-65536::
K=K*256+L-65536::
FOR T=I TO K STEP 4::
CALL PEEK(T+2,J,L)::
CALL LOAD(J*256+L-65537,0)::
NEXT T::STOP
```

RUN the program and when the cursor returns type 1 <ENTER>. Now save the program back to disk with a different name so you keep a list-table copy of the program. RUN your program again to make sure it really runs correctly. Done that, type in LIST, like the pretty screen display of course you will have to turn off to regain control, that is why it is important to save a copy with this type of protection BEFORE you try to list it. Have a think about what is happening, and all going well I will write an article next month to explain what is happening.

Finally where did I get that little trick from. I found it by doing a sector print out of a pro-gram that was protected in that manner, and reconstructing the pro-gram from the token values that where saved on the disk, using the condensed format tables printed else where in this magazine.

Been following Bob Carmany's article on programs that write programs. Well here is a full list of the condensed format codes.

CONDENSED FORMAT TABLES

DEC	HEX	MEANS	DEC	HEX	MEANS	DEC	HEX	MEANS
129	81	ELSE	171	AB	????	213	D5	LEN
130	82	::	172	AC	????	214	D6	CHR$
131	83	!	173	AD	????	215	D7	RND
132	84	IF	174	AE	????	216	D8	SEG$
133	85	GO	175	AF	????	217	D9	POS
134	86	GOTO	176	B0	THEN	218	DA	VAL
135	87	GOSUB	177	B1	TO	219	DB	STR$
136	88	RETURN	178	B2	STEP	220	DC	ASC
137	89	DEF	179	B3	,	221	DD	PI
138	8A	DIM	180	B4	;	222	DE	REC
139	8B	END	181	B5	:	223	DF	MAX
140	8C	FOR	182	B6	)	224	E0	MIN
141	8D	LET	183	B7	(	225	E1	RPT$
142	8E	BREAK	184	B8	&	226	E2	????
143	8F	UNBREAK	185	B9	????	227	E3	????
144	90	TRACE	186	BA	OR	228	E4	????
145	91	UNTRACE	187	BB	AND	229	E5	????
146	92	INPUT	188	BC	XOR	230	E6	????
147	93	DATA	189	BD	NOT	231	E7	????
148	94	RESTORE	190	BE	=	232	E8	NUMERIC
149	95	RANDOMIZE	191	BF	<	233	E9	DIGIT
150	96	NEXT	192	C0	>	234	EA	UALPHA
151	97	READ	193	C1	+	235	EB	SIZE
152	98	STOP	194	C2	-	236	EC	ALL
153	99	DELETE	195	C3	*	237	ED	USING
154	9A	REM	196	C4	/	238	EE	BEEP
155	9B	ON	197	C5	^	239	EF	ERASE
156	9C	PRINT	198	C6	????	240	F0	AT
157	9D	CALL	199	C7	quoted string	241	F1	BASE
158	9E	OPTION	200	C8	unquoted string	242	F2	????
159	9F	OPEN	201	C9	line number	243	F3	VARIABLE
160	A0	CLOSE	202	CA	EOF	244	F4	RELATIVE
161	A1	SUB	203	CB	ABS	245	F5	INTERNAL
162	A2	DISPLAY	204	CC	ATN	246	F6	SEQUENTIAL
163	A3	IMAGE	205	CD	COS	247	F7	OUTPUT
164	A4	ACCEPT	206	CE	EXP	248	F8	UPDATE
165	A5	ERROR	207	CF	INT	249	F9	APPEND
166	A6	WARNING	208	D0	LOG	250	FA	FIXED
167	A7	SUBEXIT	209	D1	SGN	251	FB	PERMANENT
168	A8	SUBEND	210	D2	SIN	252	FC	TAB
169	A9	RUN	211	D3	SQR	253	FD	# (files)
170	AA	LINPUT	212	D4	TAN	254	FE	VALIDATE

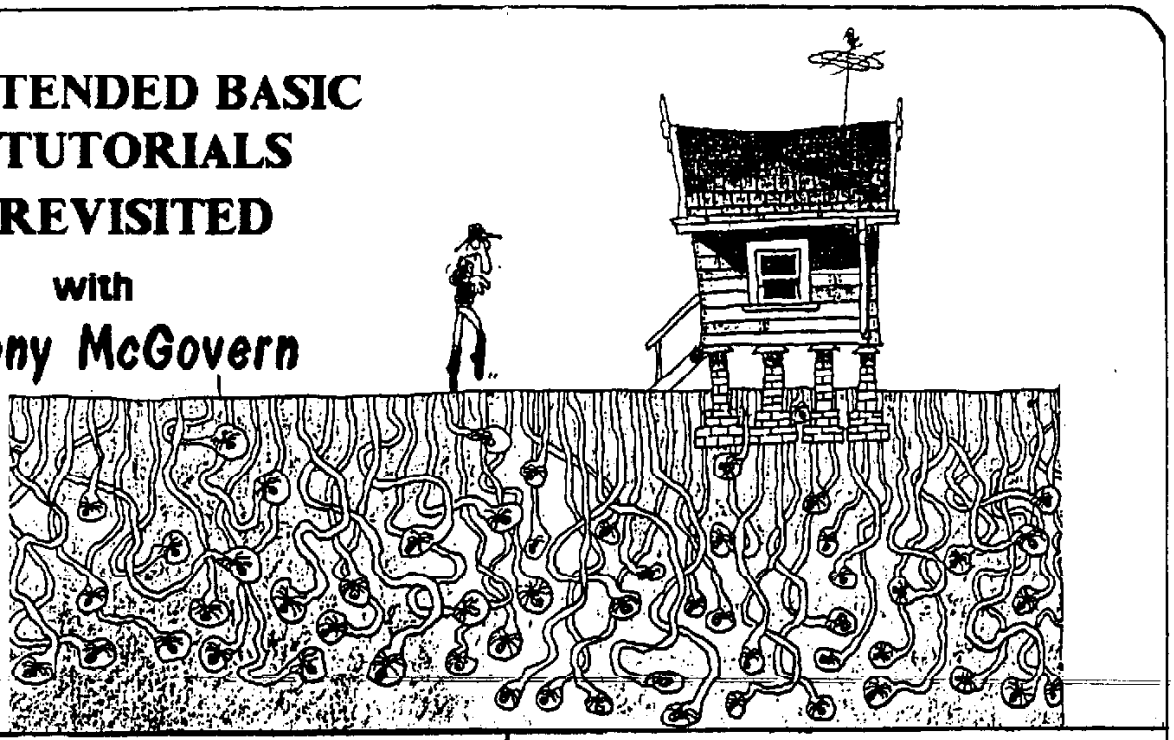
NOTE

???? means I do not know what those ascci numbers are for or if they are even used as token values. I think the list is complete without them.

EXTENDED BASIC TUTORIALS REVISITED

with

Tony McGovern



VII. ACCEPT AT and other RAMBLINGS

TI Extended Basic is a very substantial language. The XB cartridge contains 12K of ROM and 3 and a bit (the 4th one isn't full) GROMs at 6K apiece. This is on top of the 8K of console ROM and whatever parts of the 3 console GROMS are still used in XB. The tragedy of the TI-99 is that GROMS and GPL were ever invented. I guess it was TI's way of trying to keep the software market sewn up. The end result as we all know is that they shot themselves in both feet with uncanny accuracy. Instead of using the TMS9900 CRU addressing to bank switch plain ordinary ROMs or even just using GROMs only as sources of code to load into RAM (as I believe is done in the p-code card), they could have had a machine that did justice to its CPU, a real home minicomputer ..... that's all past history now.

I have been pondering on what TI should have done way back when the 99/4 was first designed, that could have been easily done at the time (or even when it was updated to the 99/4a). My conclusion is that the machine should have been given 4K of fast 16-bit CPU RAM instead of a measly 256 bytes. There would have been plenty of room with a little rearrangement and/or better decoding of memory-mapped devices (VDP, sound, speech, GROMs). This would have meant that Basic and XB system areas, sprite tables, full screen buffers, string buffers, value stack, and so on could have been in fast RAM, and even console Basic could have had full scope for character and sprite definitions (as in TI-LOGO for instance). Their cartridges could then have easily been a lot better, and let's face it, many of the earlier ones were pretty hopeless, and the later ones are all limited by lack of honest CPU RAM. The only cartridges which have stood the test of time are those that use the 32K RAM expansion. TI would then have never been dragged into that marketing war to the death (TI's that was) with that vastly inferior machine, the VIC-20. I have a suspicion that the 256 bytes happened because part of TI management wanted to protect their existing evaluation board and smaller minicomputer business.

The immediate improvement really needed in XB sub-programs is a means of examining variable values in any sub-program when program execution is halted by BREAK or errors. TI should have done it in XB by retaining the EDIT command of console Basic, allowing it to access user subprograms by name. Anyone listening out there? If so add single command array operations, full syntax checking on entry, 80 column display capability with formatting power to match,

bit-map screen functions, fast program execution and anything else will then be gravy. Then TI-99/4a owners will be most pleased to join in. The bad news is that TI is starting to cut back on support for the 9900 family despite its excellent qualities, and so it is becoming less attractive for new designs. In retrospect we still don't have these things in Basic, as the Geneve Basic remains incomplete and buggy (mid-89).

Enough ramblings and back to the tutorials ! What then is the most powerful feature in XB after SUB and CALL? A good candidate is the file system, but as this is already built into the console I will stick with commands specific to XB. The prime candidate is ACCEPT AT and its qualifying clauses (even just plain ACCEPT has some interesting improvements over INPUT but that has been treated elsewhere). This was emphasized by the recent appearance (mid-84) in a computer magazine of a long article on machine code for adding this function to IBM PC Basic (which doesn't have sub-programs either). ACCEPT AT is very useful and powerful, but has some undocumented features as well as some subtle and treacherous bugs, and is well worth talking about in this series.

The simplest level of ACCEPT AT combines the INPUT routine with its access to editing features, with cursor positioning on the display screen by the AT clause. So far this is just the input version of DISPLAY AT. The difference from INPUT is that there is no provision for prompt strings, but a DISPLAY AT soon fixes that. It also accepts input to a single variable only, and not to a whole variable list. As ACCEPT AT and DISPLAY AT do not scroll the screen, their repeated use can give a much better effect than INPUT when graphics elegance is important. Construct your own examples here or work the XB manual examples. Remember that the cursor is in XB color group 0 if you are trying to dress up the graphics.

BEEP allows an audible prompt with only one program byte (we'll talk about program length later on if this series keeps going long enough). Of course constant repetition of beeps can get a little wearing. The ERASE ALL clause provides an alternative to CALL CLEAR for clearing the screen. As compared with CALL CLEAR, ERASE ALL is slower to execute, (it seems to be line at a time) but takes less program space. Its effect is slightly different also. This little program which uses ERASE ALL with DISPLAY will make both speed and screen effects easy to see.

```
100 CALL CLEAR :: CALL COLOR
(0,3,3)
110 FOR I=1 TO 100 :: CALL C
LEAR :: NEXT I
120 FOR I=1 TO 100 :: DISPLA
Y ERASE ALL :: NEXT I
130 CALL SCREEN(11):: FOR I=
1 TO 1000 :: NEXT I
```

That's the simple pieces of ACCEPT AT -- now it starts to get interesting. VALIDATE allows the programmer to decide what characters are acceptable in a response. The computer honks (that's the word in TI-FORTH) at unacceptable inputs. Three predefined types are available. UALPHA accepts only upper-case alphabetic characters -- very useful for filenames and suchlike. This is not quite the same as depressing the alpha-lock key as it it only accepts letters, and so is incompatible with input to a numeric variable. If you are in the habit of verifying wet paint signs by touch, try that for a change. The DIGIT type does just what its name implies, and NUMERIC allows the input of any floating point number as well as plain positive integers. As with INPUT, all numbers are acceptable to a string variable, but numeric variables are fussier.

Now what if these predefined types aren't right for what you want ? Suppose only digits 1 to 4 are acceptable, as in a menu choice of 4 items labelled 1 to 4. In console Basic extra lines of code would be needed to check the input, but ACCEPT AT handles this with the clause VALIDATE("1234") or VALIDATE (I_LIKE_IT$) where the string variable has previously been set to "1234". To put it more formally, only the characters in the string argument of VALIDATE can be entered at the keyboard to be ACCEPTed.

The SIZE clause allows ACCEPT AT to be used with almost no interference to screen displays. It blanks out the specified number of characters, providing an input window of finite length, and if the length specified is negative, the characters already in the window are not erased, and form an immediate input for ACCEPTance. This is very handy for making default choices obvious to the user. Let's enter a little program to get at the essentials.

```
100 CALL CLEAR :: DISPLAY AT
(12,1):RPT$("_",28)
200 ACCEPT AT(12,2)SIZE(3):A
$
300 DISPLAY AT(15,2):A$;LEN(
A$)
400 CALL KEY(0,K,S):: IF S>0
 THEN 100 ELSE 400
```

You most likely have the Alpha-lock depressed. If so let it off, and RUN our little program. Just press ENTER the first time round, next time hit ‹space› first, and finally ‹space› first before hitting another key. This shows that ‹space›s after the last honest character entered are ignored. Try some VALIDATEs here too, if you wish. Now with the program as given, alter SIZE(3) to SIZE(-3). It now ACCEPTs whatever is in the was or is placed in that 3 character input window.

Now that's all very simple, but it brings us to the edge of the undocumented wilderness. Alter the CALL KEY(0,K,S) in the last line to CALL KEY(3,K,S) and RUN the program again, this time entering letters. Observe what happens the second time around. This answers the question of what keyboard mapping ACCEPT AT uses -- like CALL KEY(0,K,S) it uses the last one, whatever that was. Try split keyboard units in the last line. At the machine code level, a particular byte in the CPU scratchpad RAM has to be set to the key unit before calling the SCAN routine. I interpret the behaviour as showing that in the XB modules of my experience that ACCEPT AT does not alter this byte. The XB manual however does not document this behaviour at all. If XB weren't a dead language that would be a caution signal. It does need to be watched in your programs, if your last CALL KEY wasn't the key unit you want for ACCEPT AT. On the positive side you can control ACCEPT AT with a prior dummy CALL KEY to ease input for the user. An example is when a program requests input of a filename, setting the key unit to 3 makes letters come out as upper-case while still allowing other characters. Brian Rutherford of HV99 first brought the anomalous behavior to my attention and has turned up other minor undocumented variations in the use of ACCEPT AT.

Now that's not too bad, but there is worse to come. Insert a VALIDATE("123") clause in the ACCEPT AT and RUN the program. No problems there with SIZE(3), but SIZE(-3) is trickier. You can't enter invalid characters from the keyboard but unaltered "_"'s slip through. The VALIDATE appears to be exercised as characters are entered from the keyboard, and not as the edit buffer contents are transferred into the target variable. The decision to ignore trailing blanks in the input window is taken then however. Presumably a negative SIZE pre-loads the edit buffer with the screen window contents without doing a VALIDATE check. Ultimately this is not a real problem since the programmer can control what is on the screen before ACCEPT AT is invoked. Once again, the XB manual does not bind ACCEPT AT to work this way.

This behaviour does leave a weak spot in ACCEPT AT which can only be considered as a bug, but not an intractable one. Suppose you have a menu choice of items, say 1-4 by number, with default 1 pre-loaded in the SIZE(-1) window, and a VALIDATE("1234") clause to ensure proper entry for a numeric variable. What can possibly go wrong ? An evil-minded program tester would immediately delete the default using FCTN-1. An attempt to enter the blank will then cause the screen to scroll with a WARNING message. This is not a fatal error, but might as well be if your background is a carefully composed graphics screen. The workaround for this problem is not difficult, but the best one also resolves an even worse bug, so I will leave it for a little while. I do consider suppression of error trapping or warning messages by global ON ERROR

THE INFORMATION PAGE

CHRISTMAS PARTY.

Sunday 3rd December at Boolaroo Ambulance station. The party gets under way at 2:00 pm sharp. All members and their children are welcome. Bring along enough bread rolls for your family. Sausages and oinion will be supplied, Don Dorrington will be CHIEF COOK. Free drinks for the kids and plenty of games for all to enjoy. Look forward to seeing you there.

FORTH

The forth sig is on again at 7:00 pm. Richard's surgery in Whitebridge on 5 th December. All welcome. We are getting a number of our out of town members writing in expressing their interest in Forth. Hopefully we will be able to include some class notes in the newsletter starting with the January issue.
I also hope to get Richard's permission to use some of his early articles to help those just starting out with Forth.

COMMITTEE MEETING.

This, as usual will be on the second tuesday in the month 12/12/89. At Boolaroo Ambulance station again. If you havn't been to one of our end of year committee meetngs then there is a large gap in the cultural education. All welcome to attend.

EXTENDED BASIC.

Gary Jones continues with his extended basic classes on tuesday 19/12/89. Once again at Bob McClure's house in Kotara. I would suggest that you call Bob before attending.

GENERAL MEETING.

No General Meeting for december. The next General Meeting will be 23/1/90 at the Jesmond Community centre. Starts at 7:00 pm sharp.

FEBRUARY MEETING.

This is special advance notice that the Demo at the February Meeting will be done by none other than BOB CARMANY. Yes! Bob from North Carolina USA. Bob will be in Newcastle for that meeting. This is also mention in the president's column.

MEMBERSHIP.

Membership Fees for the Hunter Valley 99'ers.
Australian Membership.....$25.00
Overseas Membership.....$40.00 Australian

Adress memebrship requests or renewals to our Secretary Brian woods, his address is inside the front cover.

SOFTWARE LIBRARY.

Our young librarian Stewart Bradley is still managing the libraray inspite of his school examinations and other teenage distractions. Since the software listing was sent out with the newsletter Stewart has had an increased number of requests for software.

COMPLEMENT
****
A number that can be derived from a specified number by subtracting it from a second specified number.For example,in radix notation,the second specified number may be a given power of the radix or one less than a given power of the radix.The negative of a number is often represented by its complement.

COMPUTER
****
A data processor that can perform substantial computation, including numerous arithmetic or logic operations,without intervention by a human operator during the run.

CONDITIONAL JUMP
****
A jump that occurs if specified criteria are met.

CONTROLLER
****
Digital subsystem responsible for implementing "how" a system is to function.Not to be confused with "timing" as timing tells the system "when" to perform its function.

COUNTER
****
A circuit which counts input pulses and will give an output pulse after receiving a predetermined number of input pulses.

CRU
***
Communications Register Unit: A command-driven bit adressable I/O interface. The processor instruction can set,reset,or test any bit in the CRU array or move data between the memory and CRU data fields.

CYCLE
*****
1. An interval of space or time in which one set of events or phenomena is completed.
2. Any set of operations that is repeated regularly in the same sequence. The operations may be subject to variations on each repetition.

BIN-DAY

June 1989

REPRINTED FROM OZARK 99'ER NEWS OCT 88

APPENDIX B. DECIMAL-HEXADECIMAL - BINARY CONVERSION TABLE

Dec	Hex	Binary	Dec	Hex	Binary	Dec	Hex	Binary	Dec	Hex	Binary
0	00	0000 0000	64	40	0100 0000	128	80	1000 0000	192	C0	1100 0000
1	01	0000 0001	65	41	0100 0001	129	81	1000 0001	193	C1	1100 0001
2	02	0000 0010	66	42	0100 0010	130	82	1000 0010	194	C2	1100 0010
3	03	0000 0011	67	43	0100 0011	131	83	1000 0011	195	C3	1100 0011
4	04	0000 0100	68	44	0100 0100	132	84	1000 0100	196	C4	1100 0100
5	05	0000 0101	69	45	0100 0101	133	85	1000 0101	197	C5	1100 0101
6	06	0000 0110	70	46	0100 0110	134	86	1000 0110	198	C6	1100 0110
7	07	0000 0111	71	47	0100 0111	135	87	1000 0111	199	C7	1100 0111
8	08	0000 1000	72	48	0100 1000	136	88	1000 1000	200	C8	1100 1000
9	09	0000 1001	73	49	0100 1001	137	89	1000 1001	201	C9	1100 1001
10	0A	0000 1010	74	4A	0100 1010	138	8A	1000 1010	202	CA	1100 1010
11	0B	0000 1011	75	4B	0100 1011	139	8B	1000 1011	203	CB	1100 1011
12	0C	0000 1100	76	4C	0100 1100	140	8C	1000 1100	204	CC	1100 1100
13	0D	0000 1101	77	4D	0100 1101	141	8D	1000 1101	205	CD	1100 1101
14	0E	0000 1110	78	4E	0100 1110	142	8E	1000 1110	206	CE	1100 1110
15	0F	0000 1111	79	4F	0100 1111	143	8F	1000 1111	207	CF	1100 1111
16	10	0001 0000	80	50	0101 0000	144	90	1001 0000	208	D0	1101 0000
17	11	0001 0001	81	51	0101 0001	145	91	1001 0001	209	D1	1101 0001
18	12	0001 0010	82	52	0101 0010	146	92	1001 0010	210	D2	1101 0010
19	13	0001 0011	83	53	0101 0011	147	93	1001 0011	211	D3	1101 0011
20	14	0001 0100	84	54	0101 0100	148	94	1001 0100	212	D4	1101 0100
21	15	0001 0101	85	55	0101 0101	149	95	1001 0101	213	D5	1101 0101
22	16	0001 0110	86	56	0101 0110	150	96	1001 0110	214	D6	1101 0110
23	17	0001 0111	87	57	0101 0111	151	97	1001 0111	215	D7	1101 0111
24	18	0001 1000	88	58	0101 1000	152	98	1001 1000	216	D8	1101 1000
25	19	0001 1001	89	59	0101 1001	153	99	1001 1001	217	D9	1101 1001
26	1A	0001 1010	90	5A	0101 1010	154	9A	1001 1010	218	DA	1101 1010
27	1B	0001 1011	91	5B	0101 1011	155	9B	1001 1011	219	DB	1101 1011
28	1C	0001 1100	92	5C	0101 1100	156	9C	1001 1100	220	DC	1101 1100
29	1D	0001 1101	93	5D	0101 1101	157	9D	1001 1101	221	DD	1101 1101
30	1E	0001 1110	94	5E	0101 1110	158	9E	1001 1110	222	DE	1101 1110
31	1F	0001 1111	95	5F	0101 1111	159	9F	1001 1111	223	DF	1101 1111
32	20	0010 0000	96	60	0110 0000	160	A0	1010 0000	224	E0	1110 0000
33	21	0010 0001	97	61	0110 0001	161	A1	1010 0001	225	E1	1110 0001
34	22	0010 0010	98	62	0110 0010	162	A2	1010 0010	226	E2	1110 0010
35	23	0010 0011	99	63	0110 0011	163	A3	1010 0011	227	E3	1110 0011
36	24	0010 0100	100	64	0110 0100	164	A4	1010 0100	228	E4	1110 0100
37	25	0010 0101	101	65	0110 0101	165	A5	1010 0101	229	E5	1110 0101
38	26	0010 0110	102	66	0110 0110	166	A6	1010 0110	230	E6	1110 0110
39	27	0010 0111	103	67	0110 0111	167	A7	1010 0111	231	E7	1110 0111
40	28	0010 1000	104	68	0110 1000	168	A8	1010 1000	232	E8	1110 1000
41	29	0010 1001	105	69	0110 1001	169	A9	1010 1001	233	E9	1110 1001
42	2A	0010 1010	106	6A	0110 1010	170	AA	1010 1010	234	EA	1110 1010
43	2B	0010 1011	107	6B	0110 1011	171	AB	1010 1011	235	EB	1110 1011
44	2C	0010 1100	108	6C	0110 1100	172	AC	1010 1100	236	EC	1110 1100
45	2D	0010 1101	109	6D	0110 1101	173	AD	1010 1101	237	ED	1110 1101
46	2E	0010 1110	110	6E	0110 1110	174	AE	1010 1110	238	EE	1110 1110
47	2F	0010 1111	111	6F	0110 1111	175	AF	1010 1111	239	EF	1110 1111
48	30	0011 0000	112	70	0111 0000	176	B0	1011 0000	240	F0	1111 0000
49	31	0011 0001	113	71	0111 0001	177	B1	1011 0001	241	F1	1111 0001
50	32	0011 0010	114	72	0111 0010	178	B2	1011 0010	242	F2	1111 0010
51	33	0011 0011	115	73	0111 0011	179	B3	1011 0011	243	F3	1111 0011
52	34	0011 0100	116	74	0111 0100	180	B4	1011 0100	244	F4	1111 0100
53	35	0011 0101	117	75	0111 0101	181	B5	1011 0101	245	F5	1111 0101
54	36	0011 0110	118	76	0111 0110	182	B6	1011 0110	246	F6	1111 0110
55	37	0011 0111	119	77	0111 0111	183	B7	1011 0111	247	F7	1111 0111
56	38	0011 1000	120	78	0111 1000	184	B8	1011 1000	248	F8	1111 1000
57	39	0011 1001	121	79	0111 1001	185	B9	1011 1001	249	F9	1111 1001
58	3A	0011 1010	122	7A	0111 1010	186	BA	1011 1010	250	FA	1111 1010
59	3B	0011 1011	123	7B	0111 1011	187	BB	1011 1011	251	FB	1111 1011
60	3C	0011 1100	124	7C	0111 1100	188	BC	1011 1100	252	FC	1111 1100
61	3D	0011 1101	125	7D	0111 1101	189	BD	1011 1101	253	FD	1111 1101
62	3E	0011 1110	126	7E	0111 1110	190	BE	1011 1110	254	FE	1111 1110
63	3F	0011 1111	127	7F	0111 1111	191	BF	1011 1111	255	FF	1111 1111

DATA
****

1. A representation of facts,concepts,or instructions in a formalized manner suitable for communication,interpretation,or processing by humans or automatic means.
2. Any representations such as characters or analog quantities to which meaning is or might be assigned.

DATA BUS
**** ***

One method of input-output for a system where data are moved into or out of the digital system by way of a common bus connected to several subsystems.

DATA PROCESSING
**** **********

The execution of a systematic sequence of operations performed upon data. Synonymous with information processing.

DATA SELECTOR
**** ********

A combinational building-block that routes data from one of several inputs to a single output,according to control signals. Also called "multiplexer". Two or more such one bit selectors operating in parallel would be called a "two bit data selector," etc.

DEBUG
*****

To detect, locate, and remove mistakes from a routine or malfunctions from a computer. Synonymous with troubleshoot.

DECIMAL
****

1. Pertaining to a characteristic or property involving a selection, choice, or condition in which there are ten possibilities.
2. Pertaining to the number representation system with radix of ten.

DECIMAL DIGIT
****

In decimal notation, one of the characters 0 through 9.

DECODER
****

A conversion circuit that accepts digital input information-in the memory case, binary address information-that appears as a small number of lines and selects and activates one line of a large number of output lines.

DIGITAL
****

1. Pertaining to data in the form of digits.
2. Contrast with analog.
3. Information in discrete or quantized form; not continuous.

DIRECT ACCESS
****** ******

Pertaining to the process of obtaining data from, or placing data into, storage where the time required for such access is independent of the location of the data most recently obtained or placed in storage.

DIRECT ADDRESSING
****** **********

Method of programming that has the address pointing to the location of data or the instuction that is to be used.

DIRECT MEMORY ACCESS CHANNEL(DMA)
****** ****** ****** ************

A method of input-output for a system that uses a small processor whose sole task is that of controlling input-output. With DMA, data are moved into or out of the system without program intervention.

DOUBLE PRECISION
****** *********

Pertaining to the use of two computer words to represent a number.

DUMP
****

1. To copy the contents of all or part of a storage, usually from an internal storage into an external storage.
2. A process as in definition 1 above.
3. The data resulting from the process as in definition 1 above

DUPLEX
******

I communications, pertaining to a simultaneous two-way independent transmission in both directions. Contrast with half duplex. Synonymous with full duplex.

Mr. P. Jarmany
1504 Larson Street
Greensboro
NC 27407
USA